Binary Linear-Time Erasure Decoding for Non-Binary LDPC codes

Valentin Savin, CEA-LETI, MINATEC, Grenoble, France, valentin.savin@cea.fr

***Abstract*— In this paper, we first introduce the *extended binary representation* of non-binary codes, which corresponds to a covering graph of the bipartite graph associated with the non-binary code. Then we show that non-binary codewords correspond to binary codewords of the extended representation that further satisfy some simplex-constraint: that is, bits lying over the same symbol-node of the non-binary graph must form a codeword of a simplex code. Applied to the binary erasure channel (BEC), this description leads to a binary erasure decoding algorithm of non-binary LDPC codes, whose complexity depends linearly on the cardinality of the alphabet. We also give insights into the structure of stopping sets for non-binary LDPC codes, and discuss several aspects related to upper-layer FEC applications.**

I. Introduction

Data loss recovery – for instance, for content distribution applications or for distributed storage systems – is widely addressed using erasure codes that operate at the transport/link or the application layer of the communication system. Source data packets are extended with repair packets that are used to recover the lost data at the receiver. In this context, *Maximum Distance Separable* (MDS) codes are ideal codes, in the sense that decoding is possible as soon as the number of received packets equals the number of source data packets. However, for large block lengths, their decoding becomes untractable, and thus iteratively decoded graph-based codes constitute the main alternative. Binary *Low-Density Parity-Check* (LDPC) codes [1], with iterative decoding, have been proven to perform asymptotically close to the channel capacity [2] [3], while the decoding complexity per decoded bit is independent of the code length. Tanner represented LDPC codes by sparse bipartite graphs, and showed that they can be generalized by replacing single parity check-nodes with more general constraint-nodes [4]. Nowadays, these codes are referred as GLDPC codes and were recently investigated for the BEC [5], [6]. Another class of graph-codes, which have the attractive property of being able to generate an infinite sequence of repair packets, are the *rateless codes* proposed in [7] [8]. Over the past few years there also has been an increased interest in non-binary LDPC codes due to their enhanced correction capacity. They were mainly investigated for physical-layer channels, but at this time only few works are dealing with the BEC [9], [10], [11]. Despite their performance, non-binary LDPC codes still have to overcome the obstacle of decoding complexity in order to become attractive for practical systems.

In this paper, we introduce the *extended binary representation* of non-binary codes. From a graph point of view, the extended representation corresponds to a covering graph of the bipartite graph representing the non-binary code. The covering graph represents a binary code, and we show that any non-binary codeword can be lifted to a binary codeword of the covering graph. This gives a one-to-one correspondence between non-binary codewords and binary codewords of the covering graph that are further constrained by a simplex code[1] (that is, bits lying over the same symbol-node of the non-binary graph must form a codeword of a simplex code). By using the extended representation, we derive a binary erasure decoding for the BEC, whose complexity depends linearly on the cardinality of the alphabet, and which recover the values of the erased bits from messages received from both simplex and parity check constraints.

The paper is organized as follows. In section II we fix the notation used throughout the paper, and we review the construction of non-binary LDPC codes and their decoding over the BEC. The extended binary representation of non-binary codes is introduced in section III. In section IV we derive the binary erasure decoding of non-binary LDPC codes, and we discuss stopping sets and several aspects related to upper-layer FEC applications. Finally, section V concludes the paper.

II. Non-binary LDPC codes

We consider non-binary codes defined over an alphabet $\mathcal{A}$ with q elements, where $q = 2^p$ is a power of 2 (the last condition is only assumed for practical reasons). We assume that $\mathcal{A}$ is endowed with a vector space structure over $\mathbb{F}_2$ (the field with 2 elements), and we fix once for all an isomorphism of vector spaces:

$$\mathcal{A} \xrightarrow{\sim} \mathbb{F}_2^p \tag{1}$$

Elements of $\mathcal{A}$ will also be called *symbols*, and we say that $(x_0, \ldots, x_{p-1}) \in \mathbb{F}_2^p$ is the *binary image* of the symbol $X \in \mathcal{A}$ if they correspond to each other by the above isomorphism.

Let $\mathbb{L} = \mathcal{L}_{\mathbb{F}_2}(\mathcal{A})$ denote the algebra of $\mathbb{F}_2$-endomorphisms of $\mathcal{A}$. By evaluating elements of $\mathbb{L}$ on symbols of $\mathcal{A}$ we get a left action of $\mathbb{L}$ on $\mathcal{A}$, which will be denoted multiplicatively:

$$\mathbb{L} \times \mathcal{A} \to \mathcal{A} : \quad (h, X) \mapsto hX := h(X) \tag{2}$$

Any matrix $H \in \mathbf{M}_{M,N}(\mathbb{L})$ defines a code $\mathcal{C} \subset \mathcal{A}^N$:

$$\begin{aligned} \mathcal{C} &= \ker(H) \subset \mathcal{A}^N \\ &= \{(X_1, \ldots, X_N) \mid \sum_{n=1}^{N} h_{m,n} X_n = 0, \ \forall m = 1, \ldots, M\} \end{aligned} \tag{3}$$

This work has been partially supported by the French ANR grant N 2006 TCOM 019 (CAPRI-FEC project)

[1] A simplex code is the dual of a Hamming code.

Remark 1: Codes defined over $\mathbb{F}_q$ – the finite field with q elements – are a particular case of the above definition. The alphabet of these codes is $\mathcal{A} = \mathbb{F}_q$, whose $\mathbb{F}_2$-vector space structure is inherited from the additive operation on $\mathbb{F}_q$. Also, the internal field multiplication gives an embedding of $\mathbb{F}_q$ as a vector subspace of $\mathbb{L} = \mathcal{L}_{\mathbb{F}_2}(\mathcal{A})$. We say that *the code $\mathcal{C}$ is defined over* $\mathbb{F}_q$ if $\mathcal{C}$ is defined as the kernel of a matrix $H \in \mathbf{M}_{M,N}(\mathbb{F}_q) \subset \mathbf{M}_{M,N}(\mathbb{L})$. In this case $\mathcal{C}$ is a $\mathbb{F}_q$-vector subspace of $\mathbb{F}_q^N$.

A. The binary image of a non binary code

A sequence of symbols $(X_1, \ldots, X_N) \in \mathcal{A}^N$ may be mapped into a binary sequence of length Np via the isomorphism of (1); this binary sequence will be referred as the binary image of the given sequence of symbols. The binary images of the codewords $(X_1, \ldots, X_N) \in \mathcal{C}$ form a linear binary code $\mathcal{C}_{\text{bin}} \subseteq \mathbb{F}_2^{Np}$, called the *binary image of* $\mathcal{C}$. The isomorphism of (1) can also be used to further identify:

$$\mathbb{L} = \mathcal{L}_{\mathbb{F}_2}(\mathcal{A}) \xrightarrow{\sim} \mathcal{L}_{\mathbb{F}_2}(\mathbb{F}_2^p) = \mathbf{M}_p(\mathbb{F}_2) \tag{4}$$

Thus, by replacing each entry of $H \in \mathbf{M}_{M,N}(\mathbb{L})$ with its image under the above identification, we obtain a binary matrix $H_{\text{bin}} \in \mathbf{M}_{Mp,Np}(\mathbb{F}_2)$, which is the parity check matrix of the binary code $\mathcal{C}_{\text{bin}}$.

Remark 2: To avoid confusion, vectors will always be left-multiplied by a given matrix (unless the contrary is explicitly stated). Thus, if $h \in \mathbb{L}$ and $m_h \in \mathbf{M}_p(\mathbb{F}_2)$ is its binary image, we have $hX = Y \Leftrightarrow m_h(x_0, \ldots, x_{p-1})^t = (y_0, \ldots, y_{p-1})^t$, for all $X, Y \in \mathcal{A}$.

B. Graphical representation

The bipartite graph associated with a non-binary code $\mathcal{C}$, denoted by $\mathcal{H}$, consists of N *symbol-nodes* and M *constraint-nodes*[2] representing respectively the N columns and the M rows of the matrix H. A symbol-node and a constraint-node are connected by an edge of $\mathcal{H}$ if the corresponding entry of matrix H is a non-zero element of $\mathbb{L}$ (note that the corresponding entry is not assumed invertible!). Each edge of the graph is further labeled by the corresponding non-zero entry of H. We also denote by $\mathcal{H}(n)$ the set of constraint-nodes connected to a given symbol-node $n \in \{1, 2, \ldots, N\}$, and by $\mathcal{H}(m)$ the set of symbol-nodes connected to a given constraint-node $m \in \{1, 2, \ldots, M\}$.

C. Decoding over the BEC

In this section we assume that a non-binary LDPC code is used over the BEC(ϵ) – the binary erasure channel with erasure probability ϵ. Thus, the length N sequence of encoded symbols is mapped into its binary image of length Np, which is transmitted over the BEC; each bit from the binary image being erased with probability ϵ.

At the receiver part, the received bits are used to reconstruct the corresponding symbols of the transmitted codeword. Let n be a symbol-node of the Tanner graph. We say that a symbol $X \in \mathcal{A}$ is *eligible* for the node n, if the probability of the n^{th} transmitted symbol being X is non-zero. Tacking into consideration the channel output, the *set of eligible symbols*, denoted by $\mathscr{E}_n$, consists of the symbols whose binary images fit with the received bits (if any) of the n^{th} transmitted symbol. These sets constitute the *a priori information* of the decoder. They are iteratively updated by exchanging messages between symbol and constraint-nodes in the graph. Each message is a subset of $\mathcal{A}$, representing a set of eligible symbols, either from the constraint-node or from the symbol-node perspective:

- Each constraint-node m represents a linear combination of symbol-nodes $n \in \mathcal{H}(m)$, whose coefficients are given by the corresponding edge labels. The constraint-node m is verified if this linear combination is equal to zero. Therefore, for each $n \in \mathcal{H}(m)$ we can derive a set of eligible symbols, denoted by $\mathcal{E}_{m,n}$, according to the sets of eligible symbols $\mathcal{E}_{n'}$, with $n' \in \mathcal{H}(m) \setminus \{n\}$.
- On the other hand, each symbol-node n is involved in several linear constraints given by the nodes $m \in \mathcal{H}(n)$, all of which must be verified. Therefore, we can update the set $\mathcal{E}_n$, by tacking into account the sets of eligible symbols $\mathcal{E}_{m,n}$, with $m \in \mathcal{H}(n)$.

Using the above notation, the iterative decoding for the BEC can be expressed as follows (see also [11]):

- **constraint-node processing**
  $\mathcal{E}_{m,n} = \sum_{n' \in \mathcal{H}(m) \setminus \{n\}} h_{m,n'} \mathcal{E}_{n'}$
- **symbol-node processing**
  $\mathcal{E}_n = \mathcal{E}_n \cap \left(\bigcap_{m \in \mathcal{H}(n)} h_{m,n}^{-1} \mathcal{E}_{m,n} \right)$
  where $h_{m,n}^{-1}\mathcal{E}_{m,n} := \{X \in \mathcal{A} \mid h_{m,n}X \in \mathcal{E}_{m,n}\}$ (recall that $h_{m,n}$ is not assumed to be invertible).

These two steps are iterated as long as the cardinality of any $\mathcal{E}_n$ can be decreased. The decoding succeeds whenever all the sets of eligible symbols $\mathcal{E}_n$ get cardinality 1. It can be seen that any set of eligible symbols, $\mathcal{E}_n$ or $\mathcal{E}_{m,n}$, is a $\mathbb{F}_2$-affine subspace of $\mathcal{A}$; in particular, its cardinal is a power of 2.

Remark 3: In the above description of the erasure decoding, a symbol-node n send the same message $\mathcal{E}_n$ to all its neighbor constraint-nodes, violating the *extrinsic information principle* of a message-passing iterative decoding. However, the erasure decoding would not be changed by processing symbol-nodes in an extrinsic manner. This is due to the specificity of the BEC, which either erases a bit or transmits it correctly.

III. Extended binary representation of a non-binary LDPC code

Let $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$ denote the set of integers modulo q. The bitwise XOR operation endows $\mathbb{Z}_q$ with a vector space structure over $\mathbb{F}_2$, and the mapping $\mathbb{Z}_q \to \mathbb{F}_2^p$ that sends an integer into its binary decomposition[3] defines a vector space isomorphism.

[2]These nodes are generally called *check-nodes*. However, we will use *constraint-nodes* for non-binary codes, and *check-nodes* for binary codes.

[3]We assume that the first bit of the binary decomposition is the least significant bit

Let $h \in \mathbb{L}$ and let $m_h \in \mathbf{M}_p(\mathbb{F}_2)$ be its binary image. By using the above isomorphism, we obtain the following endomorphism of $\mathbb{Z}_q$:

$$\Phi_h : \mathbb{Z}_q \xrightarrow{\sim} \mathbb{F}_2^p \xrightarrow{{}^t m_h} \mathbb{F}_2^p \xrightarrow{\sim} \mathbb{Z}_q$$

where ${}^t m_h$ is the transpose of the matrix m_h. Thus, Φ_h satisfies $\Phi_h(i \wedge j) = \Phi_h(i) \wedge \Phi_h(j)$, where $\wedge$ is the bitwise XOR operation. The matrix $M_h \in \mathbf{M}_{q-1}(\mathbb{F}_2)$ defined by:

$$M_h(i,j) = \begin{cases} 1, & \text{if } j = \Phi_h(i) \\ 0, & \text{otherwise} \end{cases}$$

where $(i,j) \in \mathbb{Z}_q^* \times \mathbb{Z}_q^*$, is called *the extended matrix representation* of h. When h is an invertible element of $\mathbb{L}$ (or, equivalently, m_h is an invertible matrix of $\mathbf{M}_p(\mathbb{F}_2)$), Φ_h induces a permutation of $\mathbb{Z}_q^*$, thus M_h is a permutation matrix.

Remark 4: The use of $\mathbb{Z}_q$ in the above definition is only intended for indexing rows and columns of M_h by integers rather than by symbols of $\mathcal{A}$ or by elements of $\mathbb{F}_2^p$.

Example 5: Assume that $p = 3$, and let $h \in \mathbb{L}$ with binary image m_h given by:

$$m_h = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

The rows of m_h define respectively $\Phi_h(1)$, $\Phi_h(2)$, and $\Phi_h(4)$. Thus, $\Phi_h(1) = 5$ is the integer whose binary decomposition is given by the first row of m_h, and similarly $\Phi_h(2) = 7$ and $\Phi_h(4) = 6$. Finally:

- $\Phi_h(3) = \Phi_h(1) \wedge \Phi_h(2) = 2$
- $\Phi_h(5) = \Phi_h(1) \wedge \Phi_h(4) = 3$
- $\Phi_h(6) = \Phi_h(2) \wedge \Phi_h(4) = 1$
- $\Phi_h(7) = \Phi_h(1) \wedge \Phi_h(2) \wedge \Phi_h(4) = 4$

Before defining the extended binary representation a non-binary code, let us further develop this example. Consider now a non-binary code defined by a single linear constraint:

$$h_1 X + h_2 Y + h_3 Z = 0,$$

where $h_1, h_2, h_3 \in \mathbb{L}$, and $X, Y, Z \in \mathcal{A}$. Assume that after replacing h_1, h_2, h_3, and X, Y, Z by their binary images, the above equation becomes (see also Remark 2):

$$\begin{pmatrix} 1\,0\,1 \\ 1\,1\,1 \\ 0\,1\,1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0\,1\,0 \\ 0\,1\,1 \\ 1\,0\,1 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} 0\,1\,1 \\ 1\,1\,0 \\ 1\,1\,1 \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix} = 0$$

or equivalently:

$$\begin{array}{rcl} (x_0 + x_2) + y_1 + (z_1 + z_2) & = & 0 \\ (x_0 + x_1 + x_2) + (y_1 + y_2) + (z_0 + z_1) & = & 0 \\ (x_1 + x_2) + (y_0 + y_2) + (z_0 + z_1 + z_2) & = & 0 \end{array} \quad (5)$$

The main idea of the extended binary representation is to represent the code by a binary graph whose bit-nodes are in one-to-one correspondence with the set of all possible linear combinations of x_i's, y_i's, and z_i's. Therefore, we define:

$$S = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

and

$$\begin{array}{rcl} (\alpha_1, \alpha_2, \ldots, \alpha_7) & = & (x_0, x_1, x_2) \times S \\ (\beta_1, \beta_2, \ldots, \beta_7) & = & (y_0, y_1, y_2) \times S \\ (\gamma_1, \gamma_2, \ldots, \gamma_7) & = & (z_0, z_1, z_2) \times S \end{array}$$

Note that S is the parity check matrix of a Hamming code, thus $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_7)$, $\beta = (\beta_1, \beta_2, \ldots, \beta_7)$, and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_7)$ are codewords of the dual Hamming code, also called *simplex code*. The above linear equations (5) imply that:

$$M_1 \alpha + M_2 \beta + M_3 \gamma = 0,$$

where M_1, M_2, and M_3 are the extended matrices associated with h_1, h_2, and h_3. This equality corresponds to seven binary parity checks that can be represented by the binary matrix below (the zero entries do not appear in the matrix by concern of legibility). The parity checks c_1, c_2, and c_4 correspond to the linear equations of (5), and all the other parity checks (c_3, c_5, c_6, and c_7) correspond to linear combinations of the these ones.

	α							β							γ						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
c_1					1				1											1	
c_2							1						1				1				
c_3		1									1								1		
c_4						1						1									1
c_5			1											1	1						
c_6	1									1								1			
c_7				1				1								1					

Definition 6: The matrix $\overline{H}_{\text{bin}}$ of size (M(q-1), N(q-1)), obtained by replacing each coefficient h of H by its extended binary matrix M_h, is called the *extended binary matrix* associated with H. The binary code $\overline{\mathcal{C}}_{\text{bin}} = \ker(\overline{H}_{\text{bin}})$ is called the *extended binary code* associated with $\mathcal{C}$.

Definition 7: Let $S(p) \in \mathbf{M}_{p,q-1}(\mathbb{F}_2)$ be the binary matrix whose columns represent the binary decomposition of integers $j \in \{1, \ldots, q-1\}$. The *simplex code* $\mathcal{S}(p)$ is the $[q-1, p, 2^{p-1}]$ linear binary code with generator matrix $S(p)$.

Theorem 8: Let $\overline{\mathcal{C}}_{\text{bin}}$ be the extended binary code associated with a non binary code $\mathcal{C}$.
(1) Let $(X_1, \ldots, X_N) \in \mathcal{C}$, and for each $n \in \{1, \ldots, N\}$ let $(\alpha_{n,1}, \ldots, \alpha_{n,q-1}) \in \mathcal{S}(p)$ be the simplex codeword obtained by encoding the binary image $(x_{n,0}, \ldots, x_{n,p-1})$ of X_n. Then

$$(\alpha_{1,1}, \ldots, \alpha_{1,q-1}, \ldots\ldots, \alpha_{N,1}, \ldots, \alpha_{N,q-1}) \in \overline{\mathcal{C}}_{\text{bin}}$$

(2) The above mapping defines a vector space isomorphism:

$$\mathcal{C} \xrightarrow{\sim} \overline{\mathcal{C}}_{\text{bin}} \cap \mathcal{S}(p)^N$$

where $\mathcal{S}(p)^N = \mathcal{S}(p) \times \cdots \times \mathcal{S}(p) \subset \mathbb{F}_2^{N(q-1)}$ is the vector space product of N copies of $\mathcal{S}(p)$.

An intuitive interpretation of the above theorem is that a non-binary code can be represented by a graph with $N(q-1)$ bit-nodes and $M(q-1)$ check-nodes connected according to the extended binary matrix $\overline{H}_{\text{bin}}$, and N simplex-nodes connected each one to $(q-1)$ consecutive bit-nodes. Hence, within a message-passing decoding, the bit-nodes should recover their

values from messages received from both simplex and check-nodes of the graph. Although we are interested in decoding non-binary codes over the BEC, the ideas presented in this paper might be extrapolated to other channels.

Remark 9: The extended binary representation is also useful for understanding aspects related to cycles of the bipartite graph associated with a non-binary LDPC code. Assume that all the non-zero entries of H are invertible. Let $\overline{\mathcal{H}}_{\text{bin}}$ be the bipartite graph associated with the matrix $\overline{H}_{\text{bin}}$. It follows from the construction that $\overline{\mathcal{H}}_{\text{bin}}$ is a covering graph of $\mathcal{H}$, hence any cycle of $\overline{\mathcal{H}}_{\text{bin}}$ *lies over* some cycle of $\mathcal{H}$. Furthermore, let $(e_1, e_2, \ldots, e_{2\ell})$ be a cycle of length 2ℓ of $\mathcal{H}$, and let h_i denote the label of the edge e_i. Then, the number and the length of cycles of $\overline{\mathcal{H}}_{\text{bin}}$ lying over $(e_1, e_2, \ldots, e_{2\ell})$ can be derived using the cycle decomposition of the permutation Φ_h, where $h = h_1 h_2^{-1} \cdots h_{2\ell-1} h_{2\ell}^{-1}$, in a similar way as for quasi-cyclic codes (see for instance [12]).

IV. Linear time erasure decoding

Similar to section II-C, we assume that a non-binary LDPC code is used over the BEC(ϵ). Let $(X_1, X_2, \ldots, X_N)$ be the length-N sequence of encoded symbols, and let $(x_{1,0}, \ldots, x_{1,p-1}, \ldots\ldots x_{N,0}, \ldots, x_{N,p-1})$ denote its binary image of length Np, which is transmitted over the BEC; each of its bits being erased with probability ϵ. At the receiver part, the received bits are used to provide information to the corresponding bit-nodes in the extended binary graph $\overline{\mathcal{H}}_{\text{bin}}$. More precisely, for each coded symbol X_n there are $q-1$ corresponding bit-nodes in $\overline{\mathcal{H}}_{\text{bin}}$, which are denoted by $(\alpha_{n,1}, \alpha_{n,2}, \ldots, \alpha_{n,q-1})$. Recall that each $\alpha_{n,k}$ corresponds to a linear combination of $x_{n,0}, \ldots, x_{n,p-1}$, whose coefficients are given by the binary decomposition of $k \in \{1, \ldots, q-1\}$. Therefore, the bit-node $\alpha_{n,2^i}$, $0 \leq i \leq p-1$, corresponds to the bit $x_{n,i}$ from the binary sequence that is transmitted over the BEC.

The decoding algorithm is initialized as follows:

- for each received bit $x_{n,i}$ set:
$$\alpha_{n,2^i} = x_{n,i}$$
- set all the other bit-nodes $\alpha_{n,k}$ as erased

Note that a bit-node $\alpha_{n,k}$ is set at erased if either k is not a power of 2, or $k = 2^i$ but the corresponding bit $x_{n,i}$ was erased by the channel. Erased bit-nodes are then iteratively recovered as follows:

- **simplex-node processing**
for each $n \in \{1, \ldots, N\}$, if bit-nodes $\alpha_{n,k_1}, \ldots, \alpha_{n,k_i}$ are recovered (either received or recovered at the previous iterations), recover the value of $\alpha_{n,k_1 \wedge \cdots \wedge k_i}$ by:
$$\alpha_{n,k_1 \wedge \cdots \wedge k_i} = \alpha_{n,k_1} \wedge \cdots \wedge \alpha_{n,k_i}$$
- **check-node processing**
for any check-node $c \in \overline{\mathcal{H}}_{\text{bin}}$ connected to a single unrecovered bit-node $\alpha_{n,k}$, recover the value of $\alpha_{n,k}$ as the XOR of the other bit-nodes connected to c.

The simplex-node processing and the check-node processing are iterated as long as new bit-nodes $\alpha_{n,k}$ can be recovered. The decoding is successful if all the bit-nodes are recovered when it stops.

It is important to note that the above decoding is equivalent to the non-binary decoding presented in section II-C. There is a one-to-one correspondence between recovered bit-nodes and sets of eligible symbols, which can be described as follows:

- Let R_n be the set of all recovered bit-nodes $\alpha_{n,k}$ after the simplex-node processing step, for some $n \in \{1, \ldots, N\}$. Each $\alpha_{n,k} \in R_n$ gives the value of some linear combination of bits $x_{n,0}, \ldots, x_{n,p-1}$, that is:
$$\sum_{i=0}^{p-1} k_i x_{n,i} = \alpha_{n,k},$$
where $(k_0, \ldots, k_{p-1})$ is the binary decomposition of k. Let $\mathcal{E}_n \subset \mathcal{A}$ be the subset of all symbols whose binary images verify the above equation for all $\alpha_{n,k} \in R_n$. Then $\mathcal{E}_n$ coincides with the affine subspace of eligible symbols defined in section II-C. The fact that $\mathcal{E}_n$ is affine follows from the fact that for any $\alpha_{n,k}, \alpha_{n,l} \in R_n$, we also have $\alpha_{n,k \wedge l} \in R_n$.
- Let m be a constraint-node of the non-binary graph $\mathcal{H}$ and let $c_1, \ldots, c_{q-1}$ be the corresponding parity check-nodes in the binary graph $\overline{\mathcal{H}}_{\text{bin}}$. Let $n \in \mathcal{H}(m)$, and denote by $R_{m,n}$ the set of all bit-nodes $\alpha_{n,k}$ that are recovered by the check-nodes $c_1, \ldots, c_{q-1}$ from the unerased nodes among the bit-nodes $\alpha_{n',k'}$, with $n' \in \mathcal{H}(m) \setminus \{n\}$. By using the same arguments as above, $R_{m,n}$ defines a subset $\mathcal{E}_{m,n} \subset \mathcal{A}$, which coincides with the affine subspace of eligible symbols defined in section II-C. The fact that $\mathcal{E}_{m,n}$ is affine follows from the fact that whenever $\alpha_{n,k}$ and $\alpha_{n,l}$ are recovered by check-nodes c_i and c_j, then $\alpha_{n,k \wedge l}$ is also recovered by the check-node $c_{i \wedge j}$.

We discuss now the complexity of the proposed erasure decoding. The processing of each check-node is done in constant time. Since the number of check-nodes in $\overline{\mathcal{H}}_{\text{bin}}$ depends linearly on q, it follows that the check-node processing step of the decoding algorithm is done in linear time. Moreover, the simplex-node processing can also be implemented in linear time. Fix some $n \in \{1, \ldots, N\}$, and let R_{in} and R_{out} denote the sets of recovered bit-nodes $\alpha_{n,k}$ before and after the simplex-node processing. Then R_{out} is the "affine subspace" spanned by R_{in}, in the sense that $R_{\text{out}} \supseteq R_{\text{in}}$ and $\alpha_{n,k \wedge l} \in R_{\text{out}}$ for any $\alpha_{n,k}, \alpha_{n,l} \in R_{\text{out}}$, and it can be computed as follows:

$R_{\text{out}} = \{\}$
while R_{in} is not empty
  $\alpha_{n,k} \leftarrow R_{\text{in}}.\text{pop}()$
  $R_{\text{tmp}} = R_{\text{out}}$
  for $\alpha_{n,l} \in R_{\text{tmp}}$
    $\alpha_{n,k \wedge l} = \alpha_{n,k} \wedge \alpha_{n,l}$
    $R_{\text{out}} = R_{\text{out}} \cup \{\alpha_{n,k \wedge l}\}$
    $R_{\text{in}} = R_{\text{in}} \setminus \{\alpha_{n,k \wedge l}\}$
  end
  $R_{\text{out}} = R_{\text{out}} \cup \{\alpha_{n,k}\}$
end

It can be easily seen that the above implementation requires $1 + 2^1 + \cdots + 2^{|R_{\text{in}}|-1} \leq 2^p - 1 = q - 1$ computations, where $|R_{\text{in}}|$ denotes the *dimension* of the vector subspace of $\mathbb{Z}_q$ spanned by $\{k \mid \alpha_{n,k} \in R_{\text{in}}\}$.

The above discussion is resumed by the following:

Theorem 10: The complexity of the extended binary erasure decoding of non-binary LDPC codes depends linearly on the size of the alphabet.

Before concluding the paper, we would like to emphasis some other advantages of the extended binary decoding. These aspects will be developed in future works.

1) *Stopping sets*. Similar to binary LDPC codes, we can define *stoping sets*, corresponding to erasure patterns from which the decoding cannot recover. Thus, a stopping set is a subset $\mathscr{S}$ of the set of bit-nodes of $\overline{\mathcal{H}}_{\text{bin}}$, such that:

- if $\alpha_{n,k \wedge l} \in \mathscr{S}$ then either $\alpha_{n,k} \in \mathscr{S}$ or $\alpha_{n,l} \in \mathscr{S}$
- check-nodes that are neighbors of $\mathscr{S}$ are connected to $\mathscr{S}$ at least twice.

Hence, the finite length analysis of non-binary LDPC codes over the BEC can be derived by using techniques similar to those developed in [13].

2) *UL-FEC applications*. In practical systems, data packets received at the upper-layers encounter erasures, and erasure codes are used to recover the erased data packets. If non-binary LDPC codes are used in such situations, the coded symbols must be *transverse* to data packets: that is, the p bits of a symbol must belong to p different data packets (otherwise if, for instance, all the p bits of a symbol belong to the same data packet, the coded symbols will be either completely received or completely erased, and the non-binary code would operate as a binary code)[4]. The ability of the decoding algorithm of dealing with data packets instead of dealing with bits is an attractive feature of an erasure code. The proposed extended binary decoding is well-suited for UL-FEC applications as it can easily deal with data packets: the bit-nodes $\alpha_{n,k}$ would correspond to packets instead of a single bit, but the decoding would work the same way, simply by performing bitwise XOR of packets $\alpha_{n,k}$.

3) *Flexibility and small coding rates*. Another interesting feature of the proposed decoding is the possibility of using incremental redundancy in order to cope with severe channel conditions. This can be done by transmitting all the $N(q-1)$ values of the bit-nodes $\alpha_{n,k}$ over the channel, instead of transmitting only the Np bits $x_{n,i}$ of the binary image. This is illustrated in Figure 1. We use an irregular LDPC code over $\mathbb{F}_{16}$, with rate $r = 1/2$. In case that all the $N(q-1)$ values of the bit-nodes $\alpha_{n,k}$ are transmitted over the channel, the coding rate is decreased to $r' = r\dfrac{p}{q-1} = 2/15$. As it can be seen, in both situations, the code operates very close to the channel capacity. For large values of q, the incremental redundancy turns the code into an *almost rateless* code.

[4]This is contrasting with other non-binary UL-FEC codes, as the Reed-Solomon codes, for which the p bits of a symbol must belong to the same data packet.

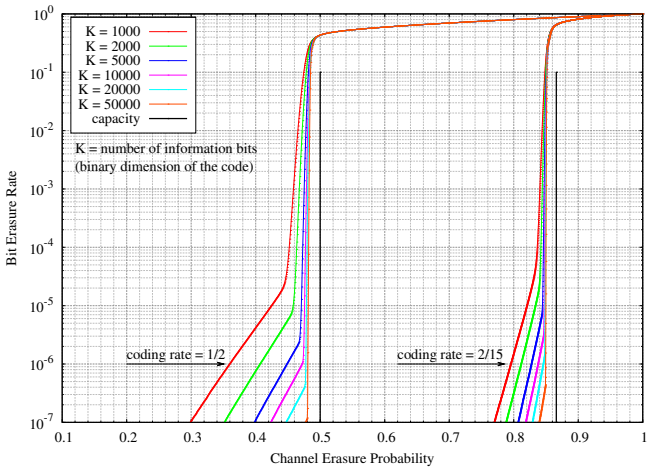

Fig. 1. Incremental redundancy using non-binary LDPC codes

V. Conclusions

We showed that non-binary LDPC codes can be described in terms of binary parity-check and simplex constraints. On the one hand, this description can be used for decoding non-binary LDPC codes, and the proposed decoding presents several attractive properties for practical applications: low complexity, capability of dealing with data packets for UL-FEC applications, on-the-fly decoding, incremental redundancy, and small coding rates. On the other hand, the proposed description gives insights into the structure of non-binary codes, and is very likely that it might be used for both finite length and asymptotical analysis of non-binary LDPC codes.

References

[1] R. G. Gallager, *Low Density Parity Check Codes*, Ph.D. thesis, MIT, Cambridge, Mass., September 1960.

[2] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, pp. 619–637, 2001.

[3] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[4] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[5] E. Paolini, M. Fossorier, and M. Chiani, "Analysis of Generalized LDPC Codes with Random Component Codes for the Binary Erasure Channel," *Int. Symp. on Information Theory and its Applications (ISITA)*, 2006.

[6] N. Miladinovic and M. Fossorier, "Generalized LDPC codes and generalized stopping sets," *IEEE Transactions on Communications*, vol. 56(2), pp. 201–212, 2008.

[7] M. Luby, "LT codes," *Proc. ACM Symp. Found. Comp. Sci.*, 2002.

[8] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52-6, pp. 2551–2567, 2006.

[9] V. Rathi and R. Urbanke, "Density Evolution, Thresholds and the Stability Condition for Non-binary LDPC Codes," *IEE Proceedings Communications*, vol. 152, no. 6, 2005.

[10] V. Rathi, "Conditional Entropy of Non-Binary LDPC Codes over the BEC," *IEEE Int. Symp. on Information Theory (ISIT)*, 2008.

[11] V. Savin, "Non binary LDPC codes over the binary erasure channel: density evolution analysis," in *IEEE Int. Symp. Applied Sciences on Biomedical and Communication Technologies (ISABEL)*, 2008.

[12] MPC Fossorier, "Quasicyclic low-density parity-check codes from circulant permutation matrices," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1788–1793, 2004.

[13] C. Di, D. Proietti, IE Telatar, TJ Richardson, and RL Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1570–1579, 2002.